Projeto de Circuitos Fotônicos Integrados

Circuitos fotônicos básicos

Atividade 3 – filtros passa-banda MZI SOI aplicados a WDM e (de-)multiplexadores

Lucivaldo Barbosa de Aguiar Junior

VIRTUS CC

Centro de Competência Embrapii em Hardware Inteligente para a Indústria

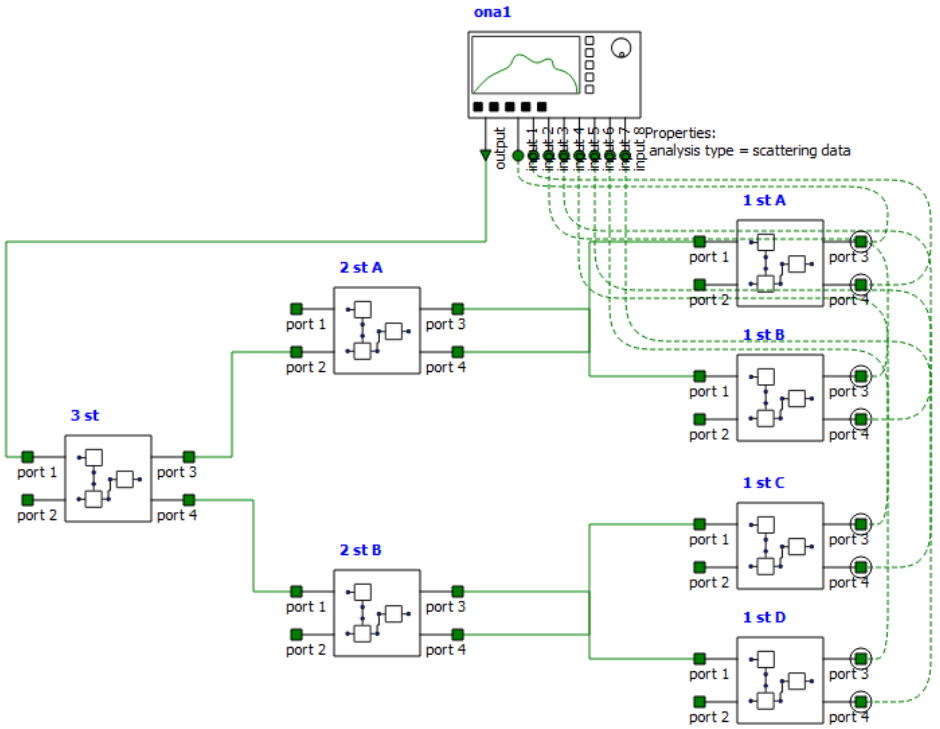
CURSOS, CAPACITAÇÃO E TREINAMENTOS

EMBRAPII

Empresa Brasileira de Pesquisa e Inovação Industrial

Sumário

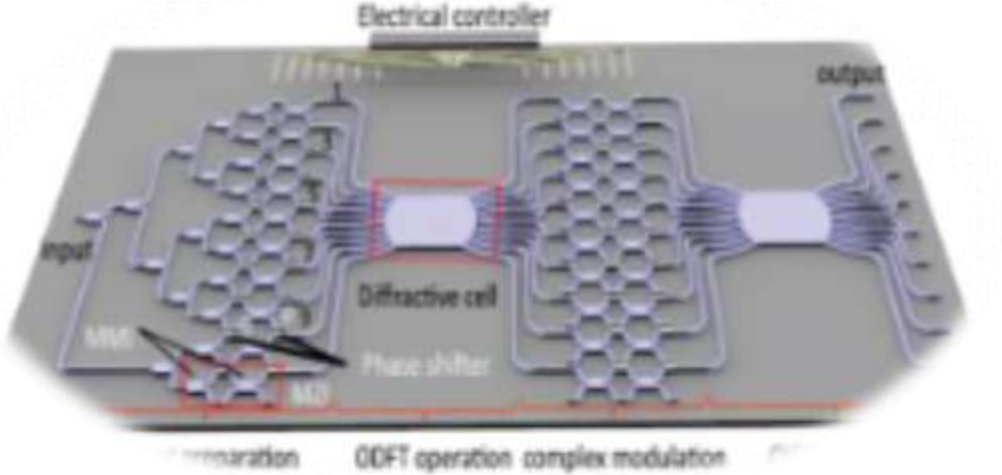
- Introdução;
- Fundamentação teórica;
- Metodologia;
- Filtro Lattice genérico via Python;
- Caso ideal;
- Filtros *SAP*, 4ª ordem e ordem mista com itens do PDK;
- Comparação entre o SAP e o utilizado;
- DRC;
- GDS;
- Referências.



Introdução

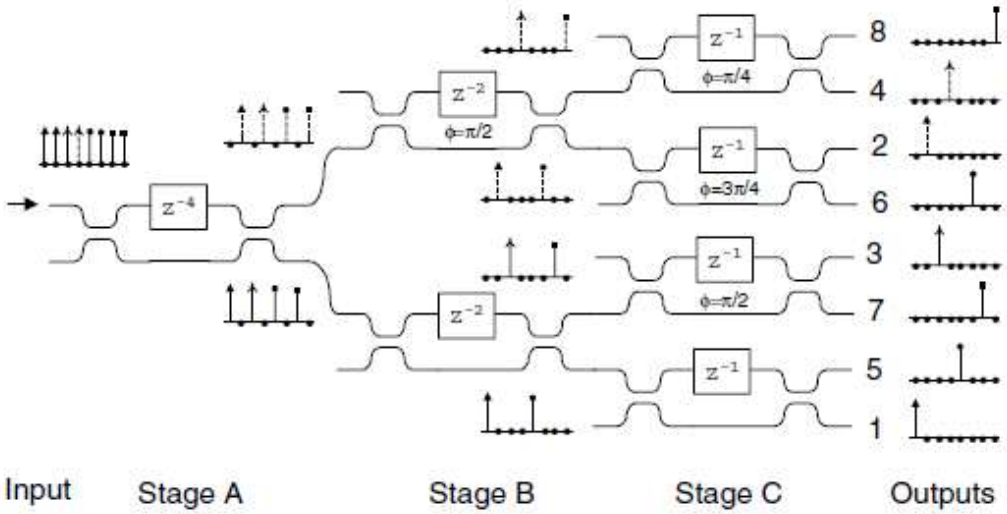
De-multiplexadores têm inúmeras aplicações em diversas áreas

- Transmissão a longas distâncias;
- Roteamento de diferentes canais para fibras;
- Computação neuromórfica;
- Redes ópticas;
- Etc.

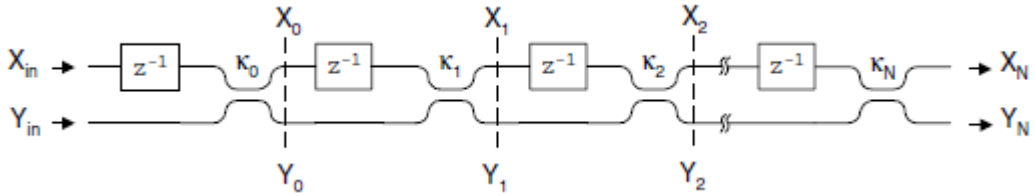


Fundamentação teórica

Entrada e saídas de demultiplexador 1:8 ideal



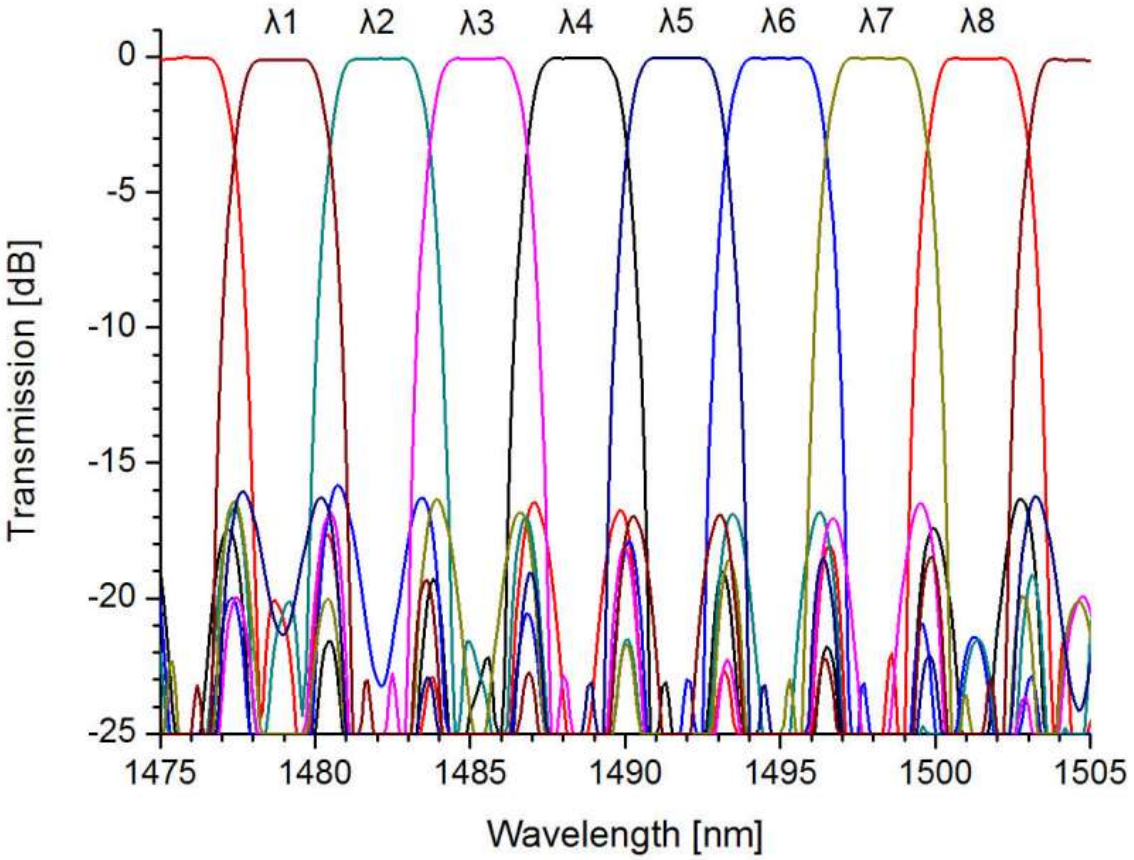
Filtro Lattice de 3ª ordem

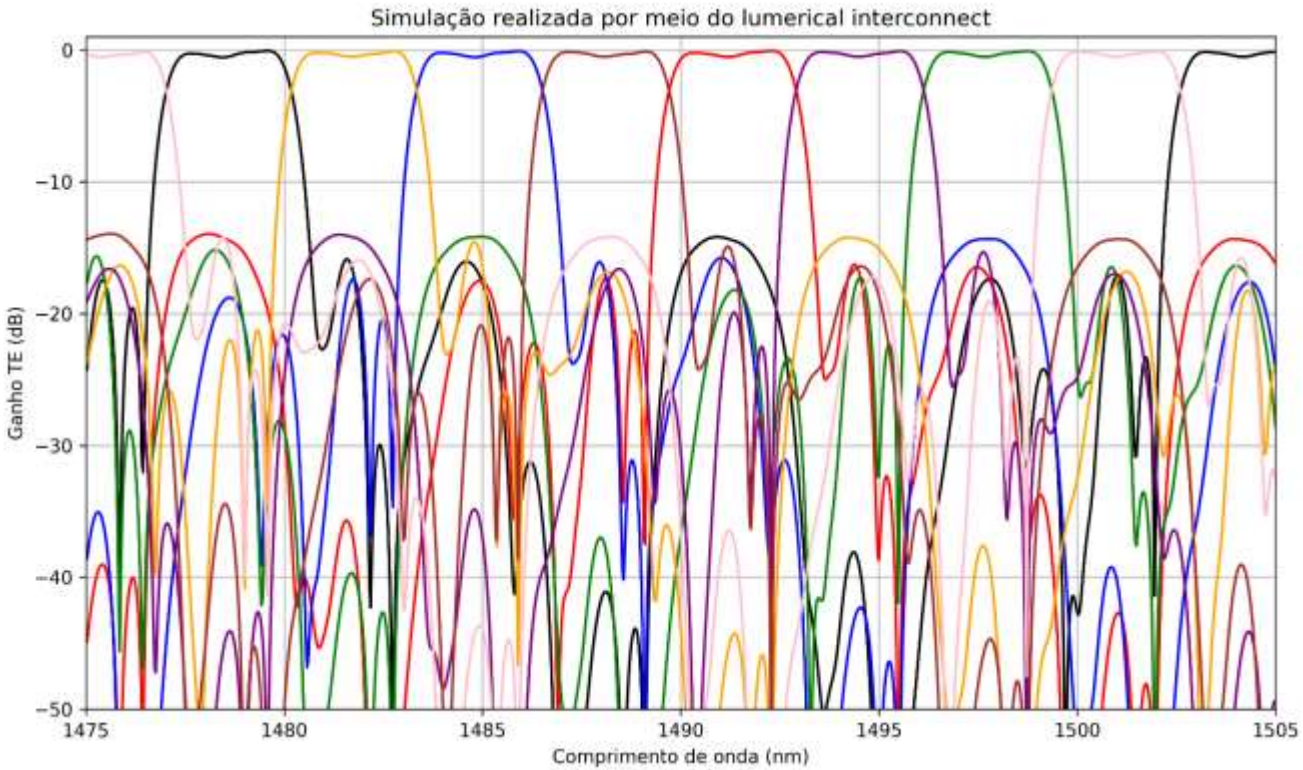


- Aumentar a ordem dos filtros aumenta a banda de passagem plana.
- **K** e Φ são obtidos numericamente por métodos recursivos.

Metodologia

Reproduzir os resultados da referência utilizada, quando possível, com os mesmos métodos e equações. Com as equações e métodos validados, são feitos os ajustes necessários ao escopo atual.



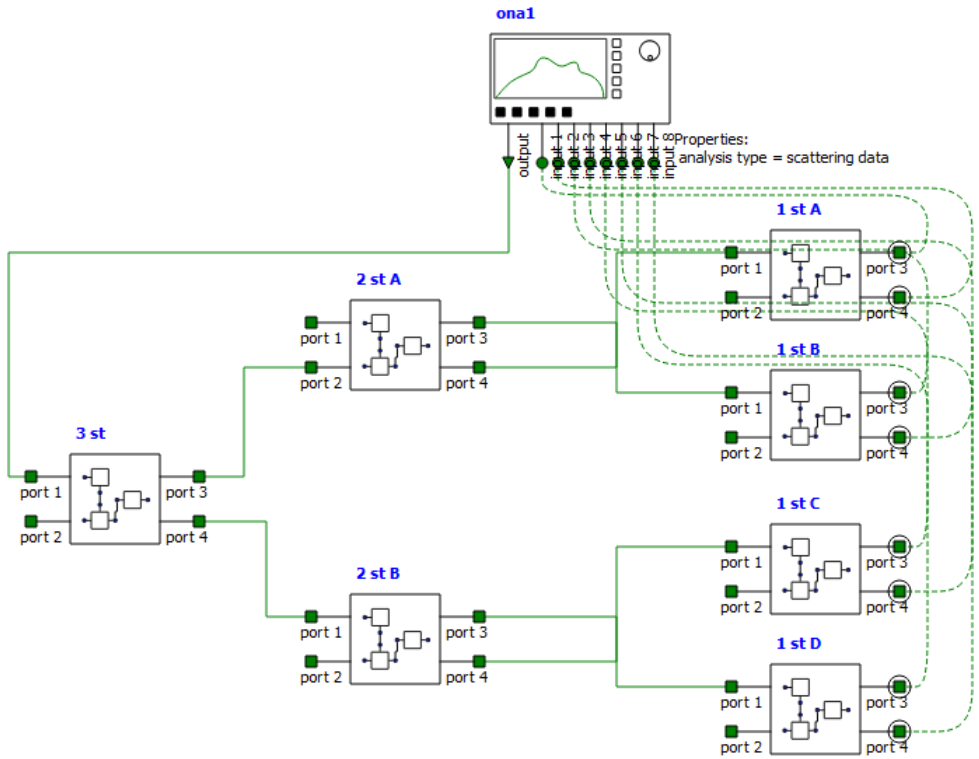


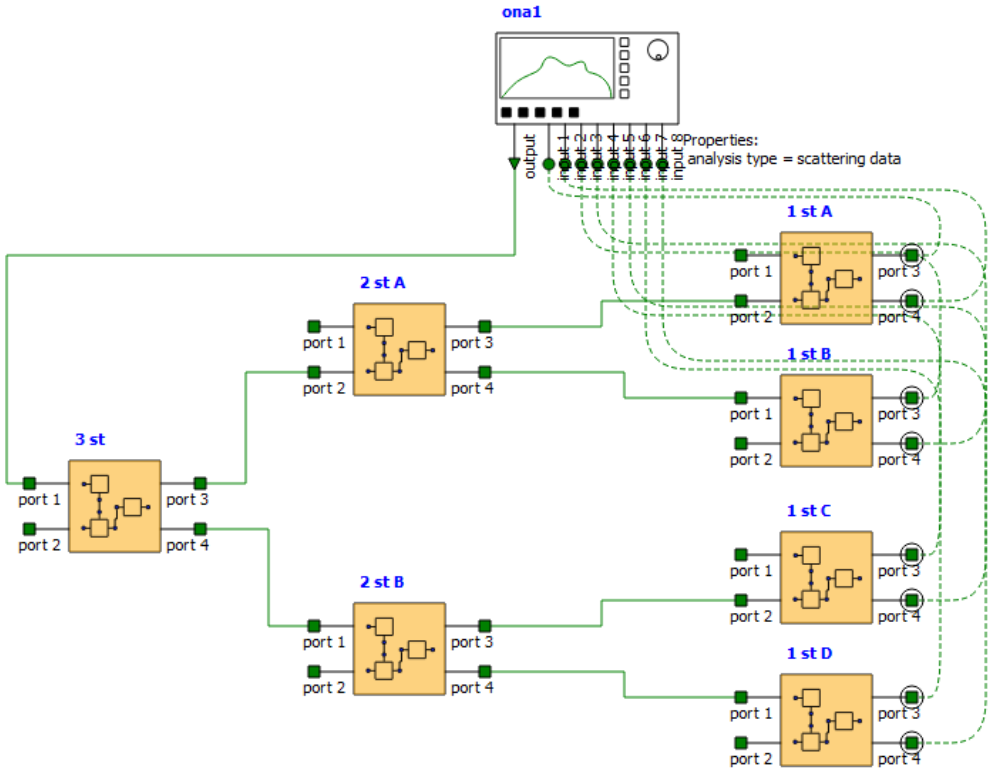
Filtro Lattice genérico

```python
def MZILatticefilter(icApi, neff, ng, L, delayLengths, k, name, nLattice):

    icApi.switchtolayout()
    # Adding and positioning the dcs and wgs
    for i in range(nLattice):
        icApi.addelement('Waveguide Coupler')
        name_dc = f'dc{i + 1}'
        icApi.set('name', name_dc)
        icApi.setposition(name_dc, 400 + 400 * i, 100)
        icApi.set('coupling coefficient 1', k[i])

    for i in range((nLattice-1)*2):
        icApi.addelement('Straight Waveguide')
        name_wg = f'wg{i + 1}'
        icApi.set('name', name_wg)
        icApi.set('effective index 1', neff)
        icApi.set('group index 1', ng)

        if i % 2 == 0:
            compr = L
        else:
            idx = i // 2
            compr = delayLengths[idx]
        icApi.set('length', compr)

        grupo = i // 2

        x = 600 + 400 * grupo

        y = 60 if i % 2 == 0 else 160

        icApi.setposition(name_wg, x, y)
```

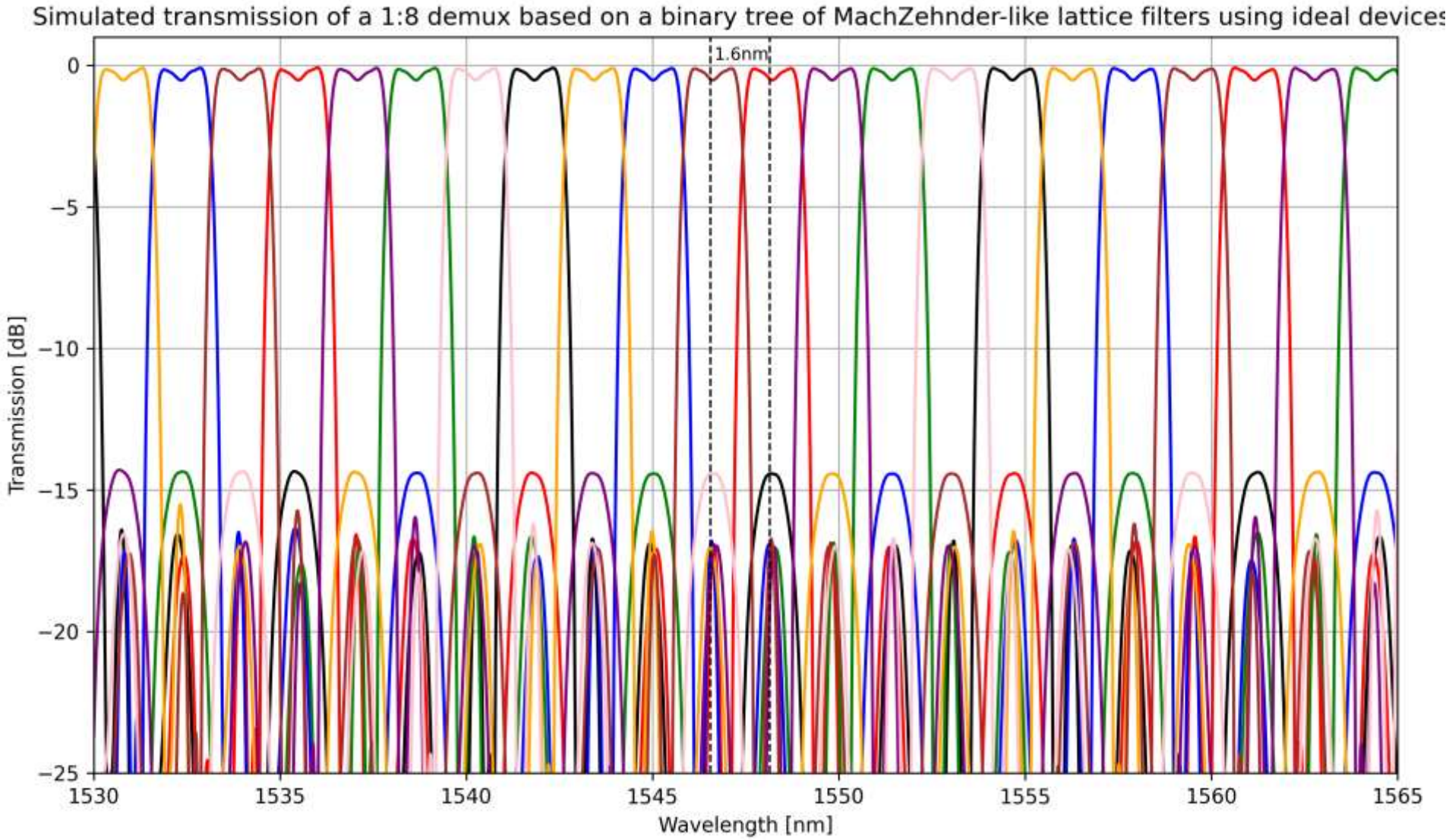
```python
# connecting em all together
for i in range(nLattice-1):
    # dcs to wgs
    icApi.connect(f'dc{i + 1}', 'port 3', f'wg{2 * i + 1}', 'port 1')
    icApi.connect(f'dc{i + 1}', 'port 4', f'wg{2 * i + 2}', 'port 1')
    # wgs to dcs
    icApi.connect(f'wg{2 * i + 1}', 'port 2', f'dc{i + 2}', 'port 1', )
    icApi.connect(f'wg{2 * i + 2}', 'port 2', f'dc{i + 2}', 'port 2', )

# criando compound element
icApi.select('dc1')
for i in range(1, nLattice+1, 1):
    icApi.shiftselect(f'dc{i}')
for j in range(0, (nLattice-1)*2, 1):
    icApi.shiftselect(f'wg{j + 1}')

icApi.createcompound()
icApi.select('COMPOUND_1')
icApi.set('name', name)
icApi.addport(name, 'port 1', 'Bidirectional', 'Optical Signal', 'Left', 0.25)
icApi.addport(name, 'port 2', 'Bidirectional', 'Optical Signal', 'Left', 0.75)

icApi.addport(name, 'port 3', 'Bidirectional', 'Optical Signal', 'Right', 0.25)
icApi.addport(name, 'port 4', 'Bidirectional', 'Optical Signal', 'Right', 0.75)

icApi.groupscope(name)
icApi.connect('RELAY_1', 'port', 'dc1', 'port 1')
icApi.connect('RELAY_2', 'port', 'dc1', 'port 2')
icApi.connect('RELAY_3', 'port', f'dc{nLattice}', 'port 3')
icApi.connect('RELAY_4', 'port', f'dc{nLattice}', 'port 4')

icApi.refresh()
return 0
```

Filtro Lattice genérico – resultado no interconnect



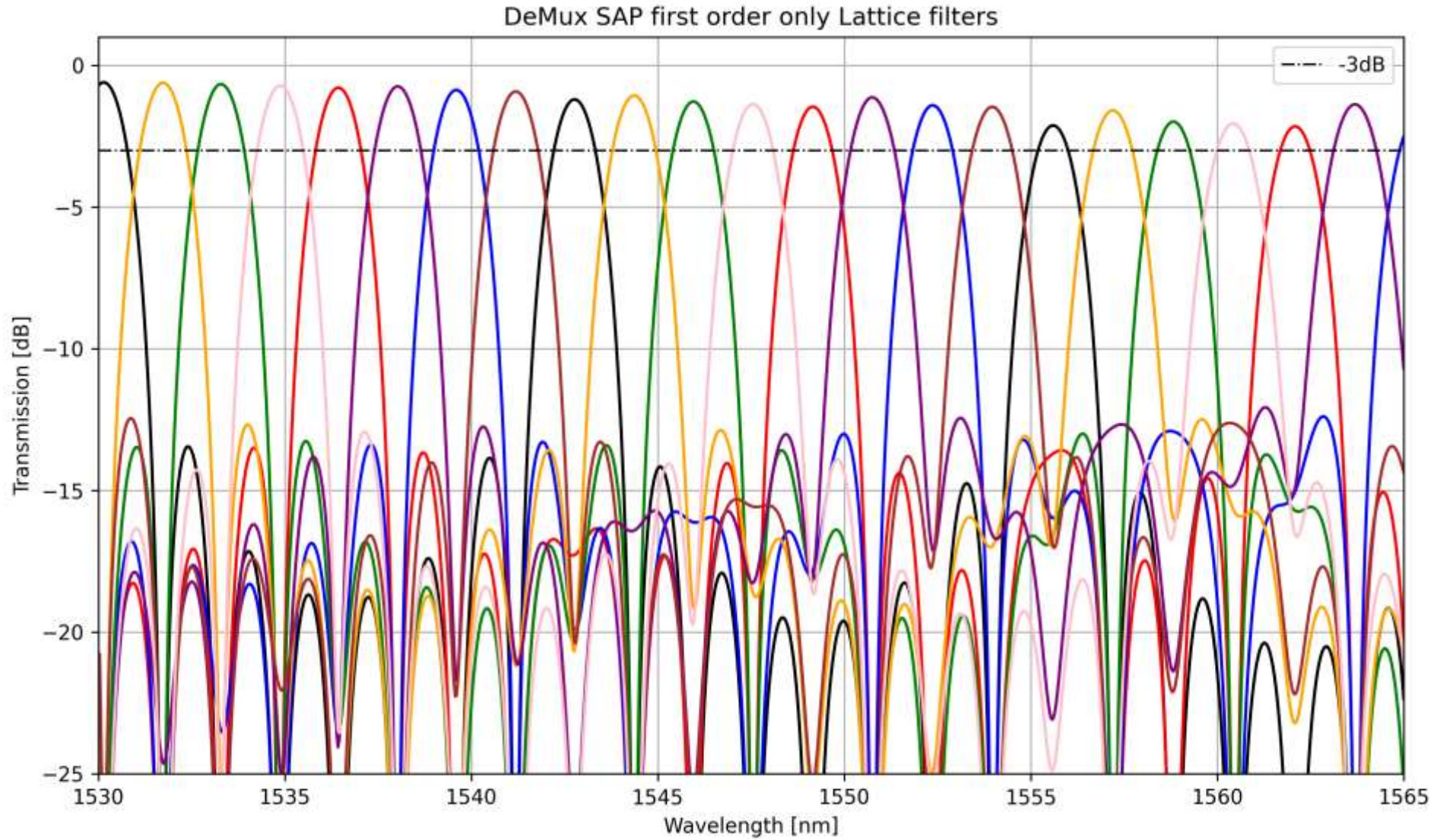


Caso ideal – gráfico de transmissão



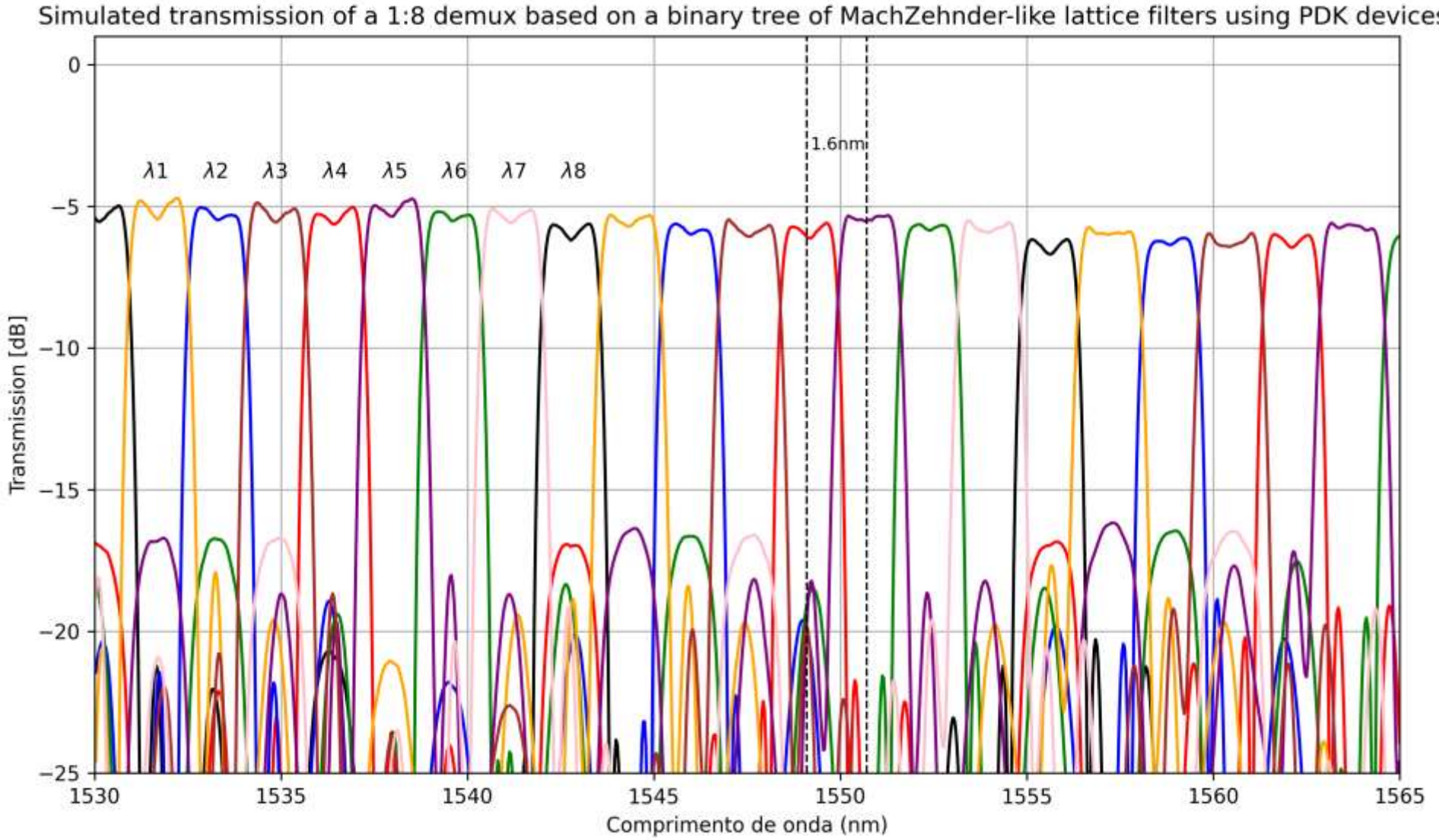
PDK SiEPIC– filtro *SAP*

- Feito utilizando apenas filtros de 1ª ordem.



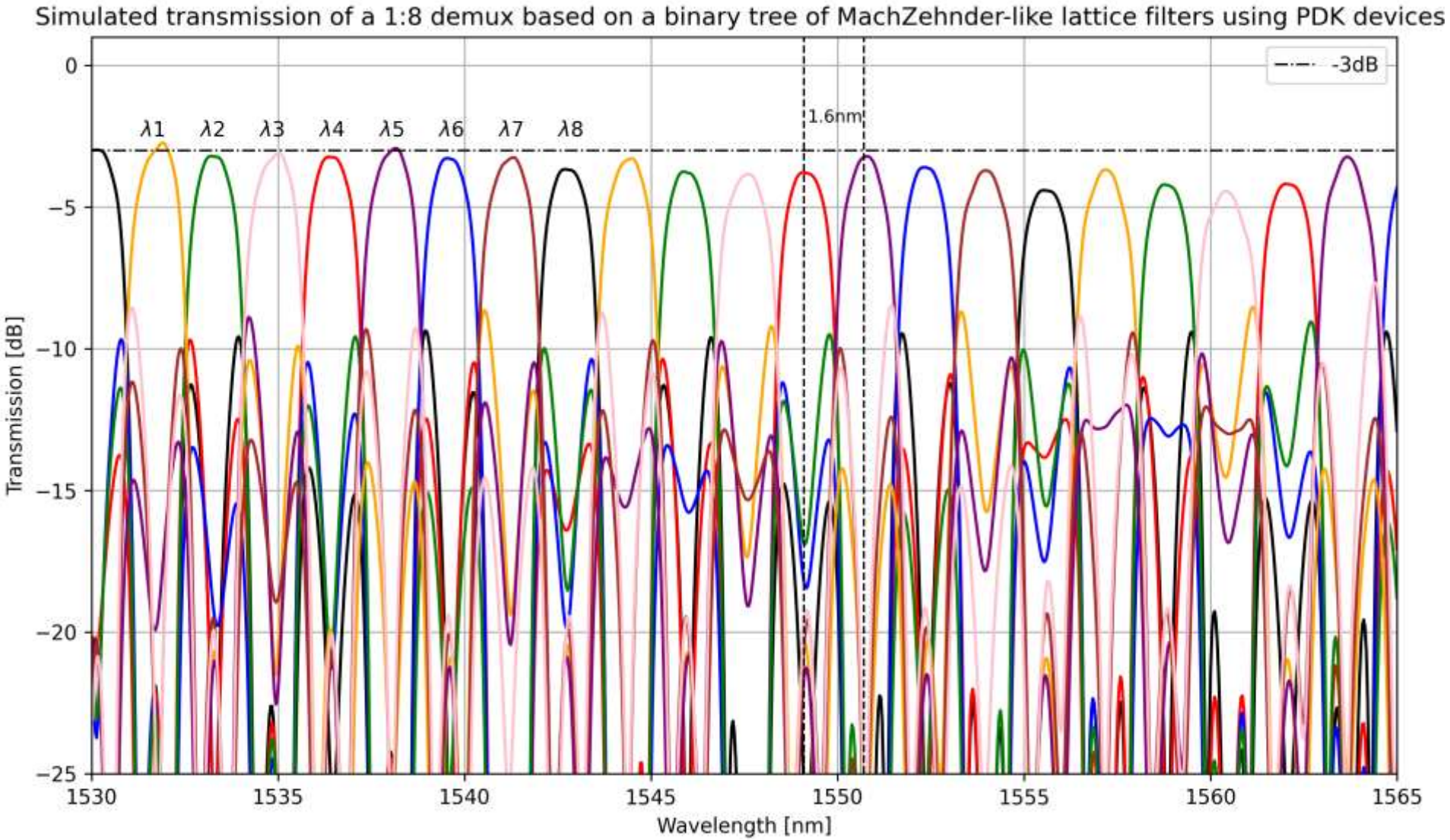
PDK SiEPIC– filtros de 4ª ordem

- Feito utilizando apenas filtros de 4ª ordem.

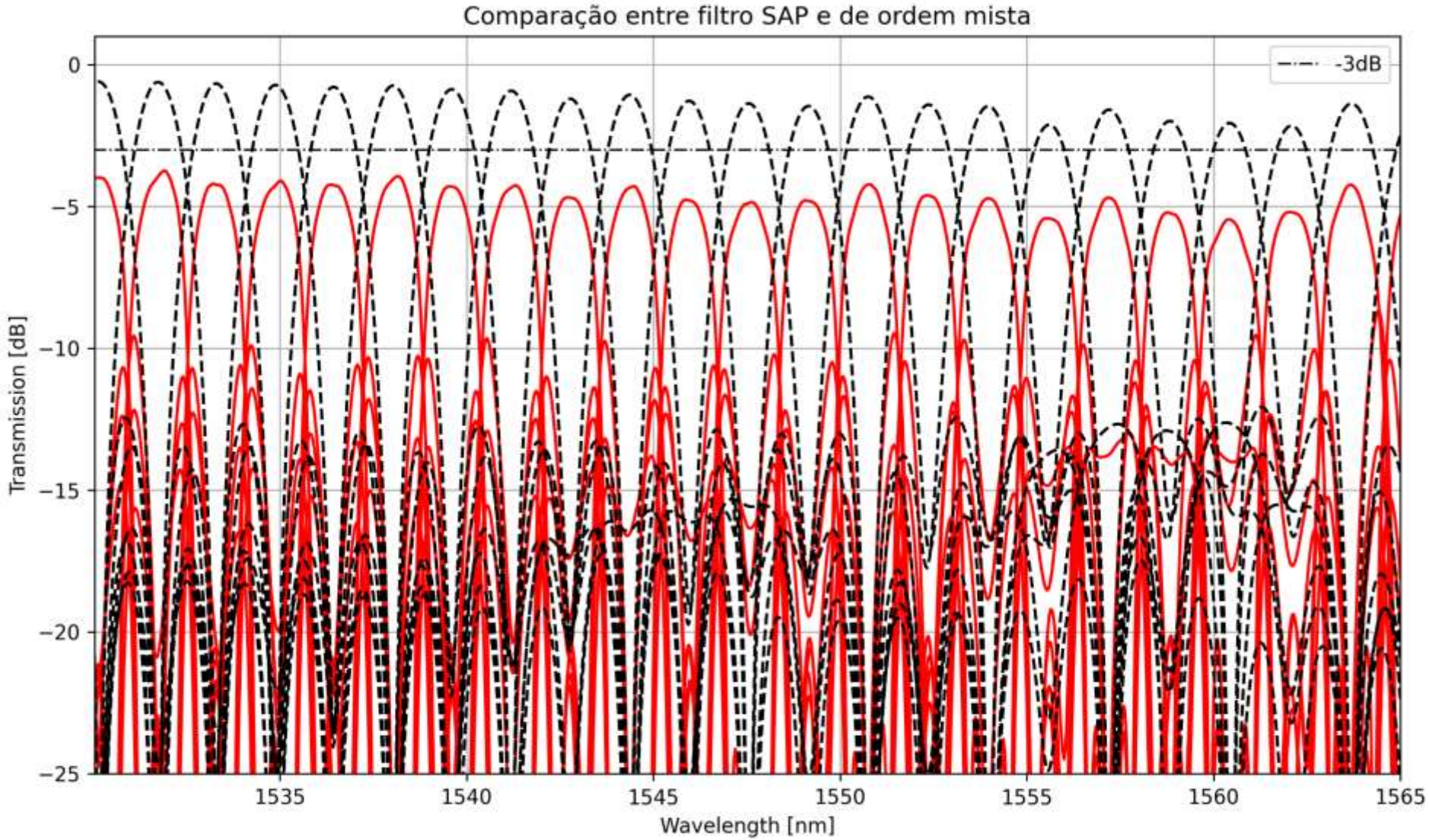


PDK SiEPIC– filtros de ordem mista

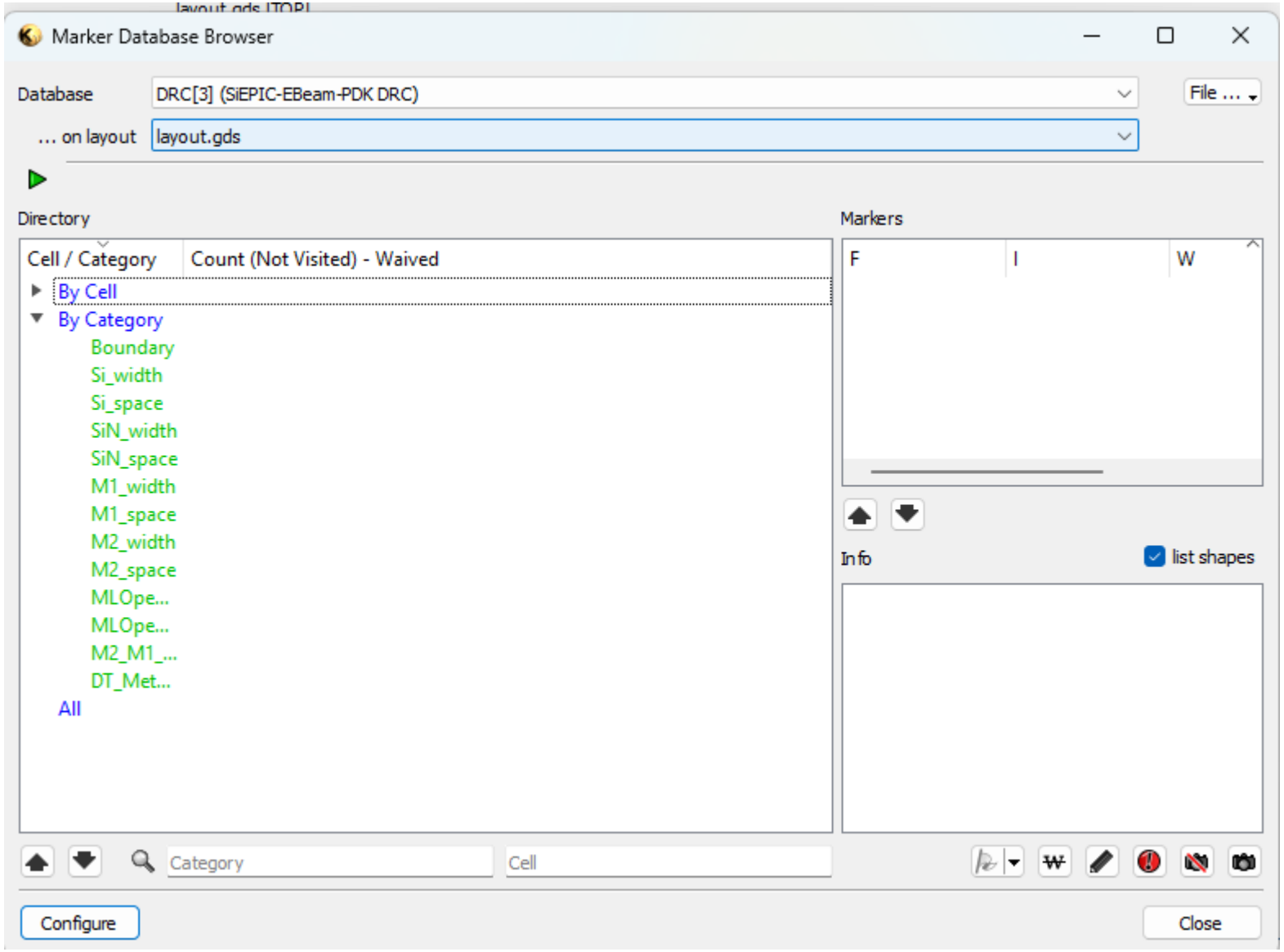
- O filtro de entrada é de 4ª ordem e o restante de 2ª ordem.

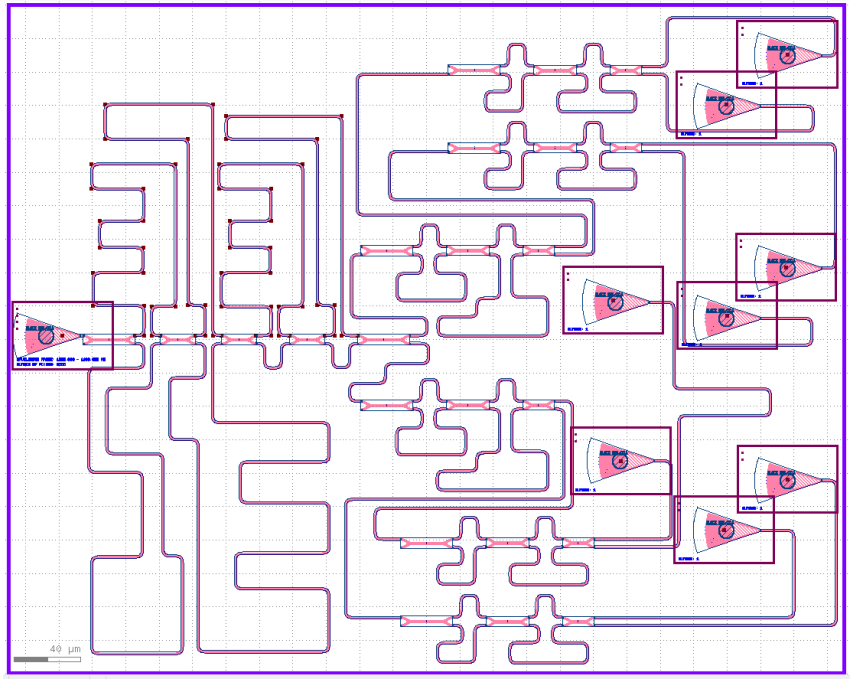


PDK SiEPIC– comparação dos filtros *SAP* e de ordem mista

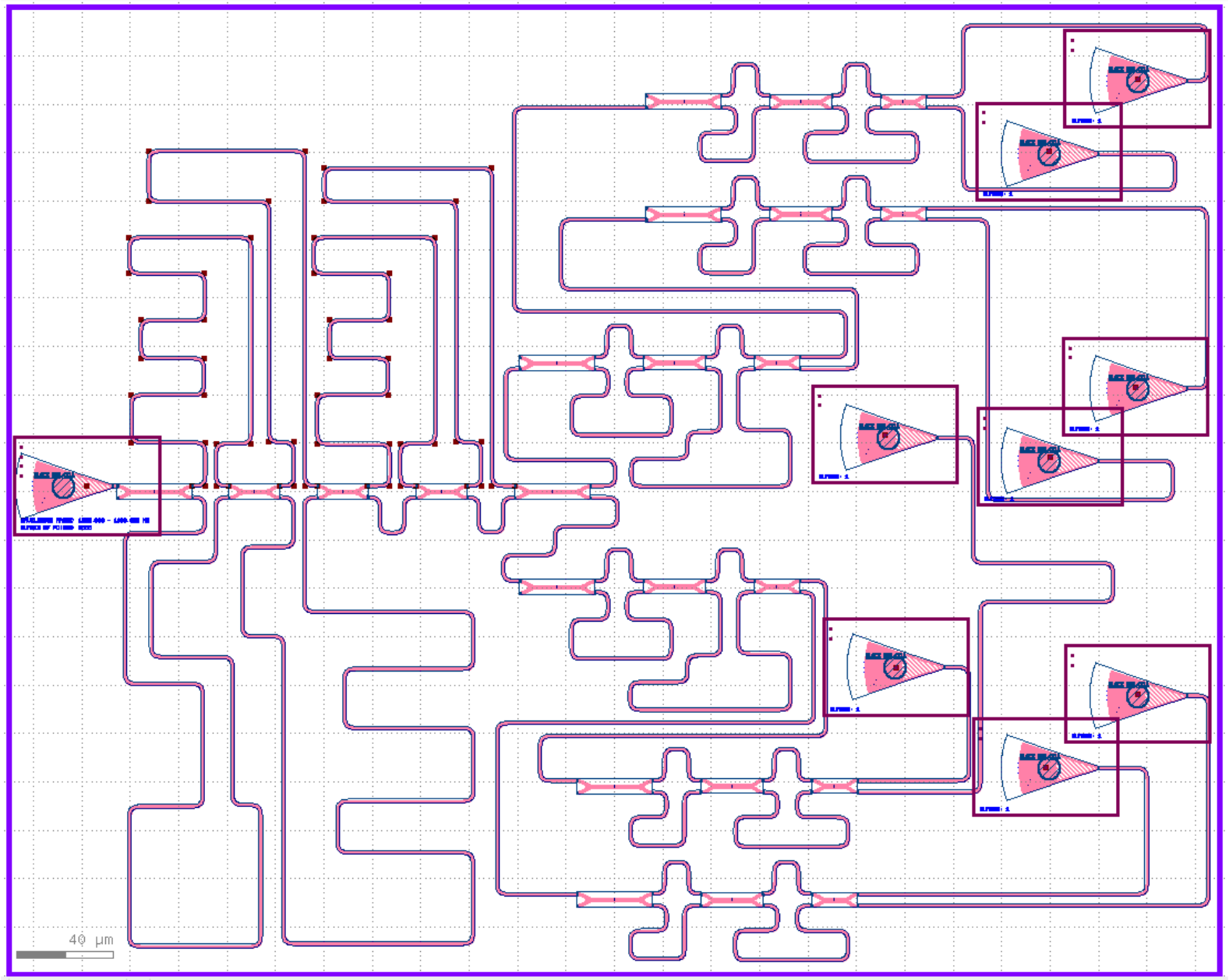


GDS– DRC


Marker Database Browser
Database
DRC[3] (SiEPIC-EBeam-PDK DRC)
File ...
... on layout
layout.gds
Directory
Markers
Cell / Category
Count (Not Visited) - Waived
F
I
W
By Cell
By Category
Boundary
Si_width
Si_space
SiN_width
SiN_space
M1_width
M1_space
M2_width
M2_space
MLOpe...
MLOpe...
M2_M1_...
DT_Met...
All
Info
list shapes
Category
Cell
Configure
Close



GDS– dispositivos do PDK SiEPIC



Referências

ANSYS. Compound Element (COMPOUND) - INTERCONNECT Element. Disponível em: https://optics.ansys.com/hc/en-us/articles/360036109554-Compound-Element-COMPOUND-INTERCONNECT-Element Aceso em: Abril de 2025.

ANSYS. Lumerical scripting language. Disponível em: https://optics.ansys.com/hc/en-us/articles/360037228834-Lumerical-scripting-language-By-category. Aceso em: Abril de 2025.

CHROSTOWSKI, Lukas; HOCHBERG, Michael E. *Silicon photonics design*. Cambridge: Cambridge University Press, 2015.

HORST, F.; GREEN, W. M. J.; ASSEFA, S.; SHANK, S. M.; VLASOV, Y. A.; OFFREIN, B. J. Cascaded Mach-Zehnder wavelength filters in silicon photonics for low loss and flat pass-band WDM (de-)multiplexing. *Optics Express*, v. 21, n. 10, p. 11652–11658, May 2013. DOI: 10.1364/OE.21.011652

OKAMOTO, Katsunari. *Fundamentals of optical waveguides*. 3rd ed. [S.l.]: Academic Press/Elsevier, 2022. ISBN 978-0-12-815601-8.

Projeto de Circuitos Fotônicos Integrados

Circuitos fotônicos básicos

Atividade 2 – filtros passa-banda MZI SOI aplicados a WDM e (de-)multiplexadores

Lucivaldo Barbosa de Aguiar Junior

VIRTUS CC

Centro de Competência Embrapii em Hardware Inteligente para a Indústria

EMBRAPII

Empresa Brasileira de Pesquisa e Inovação Industrial

virtus.ufcg.edu.br/cc